LaTeX tutorial

September 2, 2008

1 Introduction

LaTeX[1] is a powerful, free document typesetting system that is especially suited to mathematics, and is the de facto standard in the scientific publishing and academic communities. It handles gracefully formatting all manner of text as well as mathematical equations. For example:

$$\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6} \tag{1}$$

$$\begin{bmatrix} 3 & 4 \\ 2 & 7 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \tag{2}$$

$$\sqrt{2 + \sqrt{2 + \sqrt{2 + \sqrt{2 + \dots}}}} \tag{3}$$

2 Installing

The first order of business is to install the LaTeX system on your computer. Follow the instructions for the appropriate operating system.

2.1 Installing on OS X

Installing LaTeX is easy on OS X. You will just need to install the MacTeX distribution, which includes LaTeX itself as well as some convenient related

[1] By the way, LaTeX should be pronounced like "lah-tek", it should *not* be pronounced "lay-teks"! If you are talking with someone who is familiar with LaTeX and you say it "lah-tek", they will know that you are cool. Conversely, if you say "lay-teks", they will dismiss you as a hopeless n00b. Just FYI.

applications. In particular, it includes TeXShop, a program you can use to easily create and edit LaTeX documents.

To obtain MacTeX, go to `http://www.tug.org/mactex/2008/`. Download the file `MacTeX.mpkg.zip` and run the installer it contains (probably just by double-clicking on it). (As of this writing, at 7:30pm on September 2, it appears that the newest version of this file is in the process of being uploaded, so you may have to wait a bit for it to become available.) This will install a complete LaTeX system as well as some other supporting software.

`MacTeX.mpkg.zip` is rather large (1.15 GB) so if your internet connection is too slow or you don't want to wait for it to download, you can instead click on "Smaller Packages", and download and install both the mactex-additions and BasicTeX packages. This should be fine for what you need to do, and you can always install more if you ever find that there is something you need which wasn't included in these smaller packages.

2.2 Installing on Windows

You will need two pieces of software: MiKTeX is a Windows version of LaTeX itself together with many useful extension packages, and LEd is an editor you can use to create LaTeX documents. You don't *need* a special editor to create LaTeX documents; they are just plain text files so Notepad would do. However, having an editor which knows about LaTeX and has many of the needed tools already built-in saves a lot of time and annoyance.

2.2.1 MiKTeX

To download MiKTeX, go to `http://miktex.org/2.7/Setup.aspx`, scroll down, and click on the "Basic MiKTeX 2.7" installer. Once you have downloaded the installer, run it and follow the instructions to install it. If you have any questions, or problems, let me know (I don't have a Windows computer to test it on, so I'm not sure exactly what the installation process will be like). After it is finished installing, you should run the update wizard as recommended (it will probably be accessible from the Start menu).

2.2.2 LEd

To download LEd, go to `http://www.latexeditor.org/download_main.html`, download the installer, run it, and follow the instructions. Again, if you have any questions or problems, let me know.

Once LEd is installed, you should be able to use it to edit your LaTeX documents and preview the nicely typeset version. LEd has a number of additional features, such as menus from which you can select special characters and symbols to insert into your document, a built-in spell checker and thesaurus, and many other features which you will probably never need! Feel free to play around with it and see what it can do.

3 LaTeX Basics

3.1 The LaTeX philosophy

The whole idea behind LaTeX is that you should be able to specify the *content* of your document, without spending too much time worrying about *how it will look*; LaTeX has a great set of defaults for producing professional-looking documents, but also allows you to tweak whatever you like.

This separation of content and layout is very different from the way many other document processing systems work. For example, when you edit a Microsoft Word document, you see on your screen exactly how your document will look when you print it out. If you want some bold text, you select the text and click on the "bold" icon, and the text becomes bold. If you want to make a new section with a title heading, you have to type in the title, make the font bigger, and put the right amount of space around it, and so on.

With LaTeX, on the other hand, editing your document and seeing how it will look when printed are two entirely different things. When you edit your document, it is just a plain text file with some special commands to tell LaTeX how to lay out the document. If you want to see how your document will look when printed, you must run LaTeX on your document in order to produce some sort of output file (such as a PDF) which contains your nicely typeset document. (LEd can perform this step for you automatically.) For example, if you want to make some text bold, you surround it with the

command `\textbf{...}`. In order to create a new section, you simply type something like `\section{My section title}`. When you later run LaTeX on your document, you will see bold text and a section heading (with the title at a suitable size) in the output.

So, enough philosophy. Let's get started creating your very first LaTeX document. By the way, this tutorial itself is, of course, a LaTeX document! As you are reading through the rest of this tutorial, you should look at the source file `LaTeX-tutorial.tex` to see the LaTeX commands which produced the PDF you are reading.

3.2 Document layout

A basic LaTeX document looks something like this:

```
\documentclass{article}

% setup goes here

\begin{document}

  % content goes here

\end{document}
```

The first line must always have a `\documentclass` command, which specifies what kind of document you are creating. There are other document types like `book` and `report`, but for the sorts of documents you will be creating, you will never need to use anything other than `article`. Notice the syntax of LaTeX commands: a backslash indicates a command, and any parameters to the command are enclosed in curly braces, `\like{this}`.[2]

[2]By the way, if you're wondering why everywhere in `LaTeX-tutorial.tex` I write `\LaTeX\` with a backslash after it, it's because commands ignore any space that comes after them; to get a space you have to escape it with a backslash. See the difference: LaTeX with a space, LaTeXwithout a space. You very rarely need to know this, however, since it only matters for commands in the middle of some text which take no parameters; most commands you'll be using either take some parameters, or are used in math mode, where the spacing is done for you automatically.

The percent sign indicates a "comment": everything from a percent sign until the end of the line will be ignored by LaTeX, so you can use this to write notes to yourself or others that will not be included in the final output document. In this case, the comment `% setup goes here` indicates the place where various setup commands can be placed. This section before the `\begin{document}` is called the *preamble*. You'll see some examples of commands that can go in the preamble later.

Finally, the content of the document must go between `\begin{document}` and `\end{document}`.

Open up TeXShop or LEd and create a new file. Copy the above document skeleton into your blank file, and replace the "content goes here" comment with some content: for example, you could just type "My very first LaTeX document!" Then figure out how to generate a typeset PDF document as output. There will probably be some sort of button in the menu bar for doing this. Then find the generated PDF file and open it. It should just be a blank page with the text that you entered as your document content.

3.3 Bits and pieces

To make a new paragraph in a LaTeX document, just separate the paragraphs with a blank line. Otherwise, LaTeX generally ignores any extra space you put in the middle of your text, turning multiple spaces into a single space, as you can see if you take a look at the LaTeX code which generated this paragraph!

Because of the blank line above, this will be the first sentence of a new paragraph.

If you need to put something in "quotation marks," you need to use a special syntax: use two backticks for the opening quote and two apostrophes for the closing quote, like this: ``` ``quotation marks'' ```. If you use normal quotation mark characters by mistake, it will look very bad: "like this". Notice that the quotation marks in front of the word 'like' are facing the wrong way.

To make a new section in your document with the title Foo, you can type `\section{Foo}`; to make a new subsection, type `\subsection{Foo}`. The sections will be automatically numbered for you (for example, see the section headings in this document). If you don't want the sections to be numbered,

use `\section*` and `\subsection*` instead of `\section` and `\subsection`. There is also a `\subsubsection` command but most of the time you shouldn't need it.

Any automatically-numbered item (sections, equations, figures...) can be given a label; later in the document you can refer to the label and the correct number will be automatically inserted. This way, you can refer to numbered things without knowing (or caring) what number they will end up being given in the final document. In order to give something a label, put the command `\label{blah}` right after it—this will give it the label 'blah' (you can use whatever name for the label you like). Then, later, you can use `\ref{blah}` to refer to the number that label 'blah' has been given. For example, the current section is Section 3.3. As you can see if you look at the LaTeX source for this paragraph, I didn't actually type a number there—I just typed `\ref{sec:bits}` to refer to the label which I earlier gave to this section.

3.4 Special environments

Environments are enclosed in `\begin{foo}` ... `\end{foo}` pairs (where 'foo' is the name of the environment), and can specify some special way to typeset their contents. For example, you have already seen that the entire document contents must be enclosed in a `document` environment. There are a few other special environments you should be aware of.

The `itemize` environment lets you make a bulleted list of items:

- Like this.
- Each item inside the environment should be preceded by the special `\item` command.
- For an example, look at the LaTeX source that was used to produce this list.

The `enumerate` environment lets you make a numbered list of items:

1. Like this.

2. Again, each item should be preceded by the `\item` command.

3. The numbers are inserted automatically, so you can add, delete, or move items around without worrying about the numbering getting messed up!

   (a) You can even have nested `enumerate` environments.

   (b) Like this.

4. One more item.

3.5 Mathematics

To include a mathematical expression in the middle of some text, enclose the mathematics in dollar signs. For example, typing `$3+x=9$` produces $3+x = 9$. Notice how bad it looks if you don't use dollar signs: 3+x=9.

In mathematics mode (anything inside dollar signs) there are also a huge number of commands you can use to produce special mathematical symbols.

- You can make superscripts using the `^` (carat) character. For example, `$x^2$` produces x^2. If the superscript consists of more than one character, be sure to enclose the superscript portion in curly braces so LaTeX knows what should be included in the superscript. For example, compare `$x^i+2$`, which produces $x^i + 2$, with `$x^{i+2}$`, which produces x^{i+2}.

- You can make subscripts using `_` (underscore). For example, `$x_3$` looks like this: x_3.

- You can make a square root using the `\sqrt{...}` command. For example, `$\sqrt{x+2}$` looks like this: $\sqrt{x+2}$.

- You can make fractions using the `\frac{...}{...}` command. For example, `$\frac{y+2}{5}$` looks like this: $\frac{y+2}{5}$.

- Some mathematical symbols can be typed directly from your keyboard, like $=$, $+$, $-$, $>$, and $<$. However, there are a very large number of special mathematical symbols that do not correspond to a key on

the keyboard but can be produced using a special LaTeX command. For example, ∞ (`\infty`), $\to$ (`\to`), $\geq$ (`\geq`), $\leq$ (`\leq`), $\neq$ (`\neq`), $\cdot$ (`\cdot`), and π (`\pi`). There are many others, but you will learn them as we go along.

Sometimes you will want to typeset an equation, such as

$$x^3 + 15 = 33$$

by itself on a separate line, instead of in the middle of a sentence like $x^3+15 = 33$. To achieve this, you can just enclose the equation in a `\[ ... \]` pair, instead of dollar signs. Sometimes you also want to give an equation a number so you can refer to it later, like this:

$$x^3 + 15 = 33 \tag{4}$$

We can solve equation (4) to find that $x = \sqrt[3]{18}$. To achieve this, enclose the equation in an `equation` environment. Of course, you can give equations `\label`s just like sections. To refer to equation numbers with the parentheses included (as I did above), you can use `\eqref` instead of just `\ref`.

3.6 Making your own commands

One final thing before you will know enough LaTeX to get started. There is a very helpful feature which lets you define your own commands! Let's say you find yourself typing "my helicopter is full of eels" a lot, and it's getting kind of annoying to type it out every time. You can define a new command to generate this text, like this:

```
\newcommand{\mhfe}{my helicopter is full of eels}
```

Any `\newcommand`s should go in the preamble of your document, that is, *before* the `\begin{document}`.

Once you have defined this command, you can just type `\mhfe` instead of the entire phrase, and the phrase will be included in the output document: my helicopter is full of eels.

You can even create commands which have missing details that you can fill in later by providing parameters to the command. For example, suppose you find yourself often writing things like $\sqrt{\sqrt{\sqrt{\sqrt{3}}}}$ and $\sqrt{\sqrt{\sqrt{\sqrt{5}}}}$. It's annoying to have to type `\sqrt{\sqrt{\sqrt{\sqrt{...}}}}` every time, so you want to create a special command to save you some typing. The problem, of course, is that you might want to change the number inside the square root each time. The solution is to define a command like this:

```
\newcommand{\ssss}[1]{\sqrt{\sqrt{\sqrt{\sqrt{#1}}}}}
```

The `[1]` in square brackets says how many parameters the command needs. The `#1` refers to whatever parameter is provided. For example, `\ssss{\pi}` automatically turns into `\sqrt{\sqrt{\sqrt{\sqrt{\pi}}}}`, thus producing $\sqrt{\sqrt{\sqrt{\sqrt{\pi}}}}$. Of course, you could also make a command with multiple parameters and use `#1`, `#2`, and so on to access them.

3.7 `precalc.sty`

I have created a special package file, `precalc.sty`, which contains a number of commands and settings you can use while preparing your solutions. For now, simply place a copy of `precalc.sty` in the same directory with any `.tex` file you create; then you can use it by putting `\usepackage{precalc}` in the preamble of your `.tex` file. The solution template file already contains `\usepackage{precalc}`, so you really don't have to worry about it beyond making sure you have a copy of it in your working directory.

Even better, there is probably a way to "install" `precalc.sty` with MiKTeX so that you can use it automatically, without having to put a copy in your working directory. I don't know how to do this with MiKTeX, but once you have MiKTeX installed we could try to figure it out.

4 Learning more

If you want to learn more about LaTeX and the underlying typesetting system TeX beyond what is covered in this tutorial, a great starting place

is to read *The Not-So-Short Introduction to LaTeX2e*, `http://tug.ctan.org/tex-archive/info/lshort/english/lshort.pdf`. There is also much information which can be found at `http://www.latex-project.org/` and `http://ctan.org`.